TIME-SERIES PREDICTION USING CONSTRAINED FORMULATION FOR NEURAL NETWORK TRAINING AND CROSS VALIDATION

Benjamin W. Wah and Minglun Qian

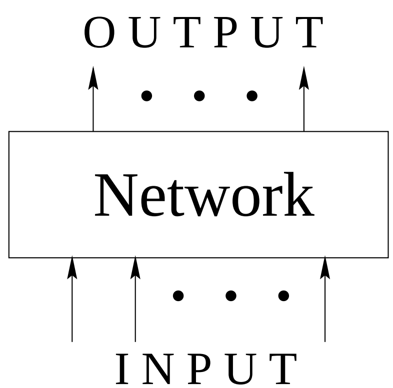
Dept. of Electrical and Computer Engineering
and the Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
http://manip.crhc.uiuc.edu

Outline

- **Motivations**
- **Constrained formulations for artificial neural-network (ANN) training**
- **Training algorithms for constrained formulation**
  - Discrete Lagrange-multiplier theory (DLM)
  - Constrained simulated annealing (CSA)
  - Integration of back-propagation and CSA
- **Experimental results**
- **Conclusions**

ANN Model for Time-Series Prediction

- Time-series prediction
  - Given a sequence of values observed in the past, predict future values
- ANN model for time-series
  - Feedforward
  - Weights adjustment



Traditional Formulations for ANN Training

- Unconstrained formulation

$$\min_{w} \quad E(w) = \sum_{t=1}^{n} ||\vec{o}_t(w) - \vec{d}_t||^2, \tag{1}$$

- Training algorithms
  - BP/BP variants and gradient-based methods
  - Genetic algorithms
  - Simulated annealing
- Issues
  - No guidance when search reaches a local minimum of $E(w)$
  - Nonuniform errors on patterns – not best for prediction

Traditional Cross-Validation

- Divide historical data into two disjoint sets
  - Training set
  - Cross-validation set
- Issues
  - Hard to choose appropriate validation set: where and how long?
  - Data used for cross-validation cannot be used for training
  - Only one validation set is used at any time: not good when time series is multi-stationary

Motivations

- Motivations
  - Unconstrained formulation leads to either poor solution quality or long training time
  - Multiple cross-validation sets are needed for multi-stationary time-series
  - Unsatisfied pattern may provide extra guidance for further search
  - Our previous successful application of constrained formulation
    - * Two-spiral problem: 4 hidden units with only 19 weights
- Proposed solution
  - Use constrained formulation for ANN training and cross validation
  - Solve constrained problem using constrained simulated annealing

Performance Metrics

- One output unit
- Normalized mean square error (nMSE)

$$\varepsilon = \frac{1}{\sigma^2 N} \sum_{t=t_0}^{t_1} (y(t) - d(t))^2, \qquad (2)$$

  - σ^2 is the variance of the true time series in $[t_0, t_1]$
  - $y(t)$ is the actual output, $d(t)$ is the desired output
  - N is number of patterns in the measurement

- Open-loop single-step measurement: external input is true observed data
- Close-loop iterative measurement: external input is predicted output obtained in the last iteration

Proposed Constrained Formulation

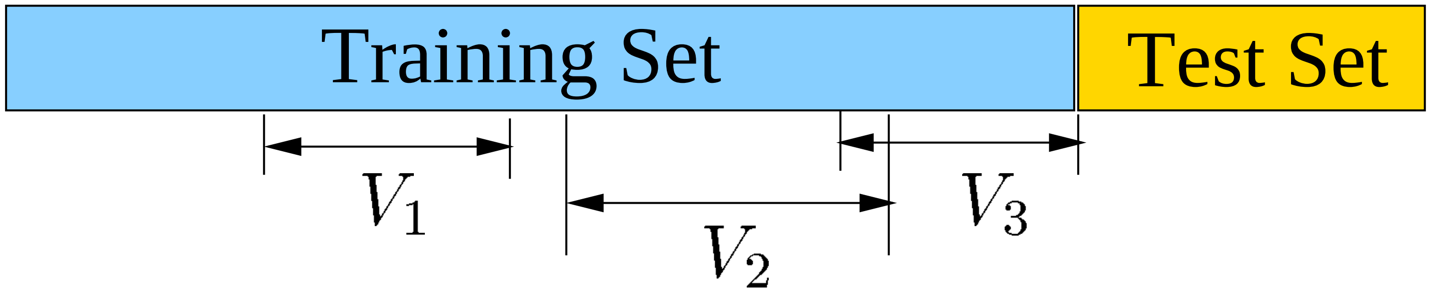
- Each pattern treated as an additional constraint
- Constrained formulation

$$\begin{aligned} \min_w \; & E(w) = \sum_{t=1}^{n} max^2\{|o_t(w) - d_t| - \tau, 0\} \\ \text{s.t.} \; & h_t(w) = |o_t(w) - d_t| \le \tau, \end{aligned} \quad (3)$$

  - τ decreases towards 0 as looser constraints are satisfied
  - Equivalent to unconstrained formulation when $\tau = 0$
- Advantages
  - Unsatisfied constraints provide extra guidance when search reaches a sub-optimum of the objective function
  - Introduction of τ allows even training for all patterns: important when there is no solution to $E(w) = 0$

Proposed Cross-Validation Method

- Multiple validation set(s) within training set
- Iterative and single-step validation errors added as new constraints
- Advantages
  - Training patterns fully used
  - Multiple validation sets cover multiple regimes in a multi-stationary time series
  - Flexibility in choosing validation sets – location and length



Constrained Formulation with Cross Validation

- Constrained formulation

$$
\begin{array}{lll}
\min_w & E(w) = \sum_{t=1}^{n} max^2\{|o_t(w) - d_t| - \tau, 0\} & \\
\text{s.t.} & h_t(w) = |o_t(w) - d_t| \leq \tau & (4) \\
 & h_i^I(w) = \varepsilon_i^I \leq \tau_i^I & \\
 & h_i^S(w) = \varepsilon_i^S \leq \tau_i^S, &
\end{array}
$$

  - ε_i^I: nMSE of iterative validation error on the i^{th} validation set
  - ε_i^S: nMSE of single-step validation error on the i^{th} validation set
  - τ_i^I and τ_i^S: small positive values and refined successively as training progresses

- Constrained formulation solved by *constrained simulated annealing* (CSA) which is based on *discrete Lagrange-multiplier theory*

Discrete Lagrange-Multiplier Theory

- Discrete equality-constrained *nonlinear programming problem* (NLP):

$$\begin{aligned} \text{minimize}_x \quad & f(x) \\ \text{subject to} \quad & h(x) = 0, \end{aligned} \tag{5}$$

where $x = (x_1, \cdots, x_n)$ is discrete.

- Inequality constraint transformation: $g_j(x) < 0 \Longrightarrow max(g_j(x), 0) = 0$
- Generalized discrete augmented Lagrangian function of (5):

$$L_d(x, \lambda) = f(x) + \lambda^T H\left(h(x)\right) + \frac{1}{2}||h(x)||^2, \tag{6}$$

where H is a non-negative transformation

- $Neighborhood$ $\mathcal{N}(x)$ of x
  - *Finite* user-defined set of discrete points that satisfy reachability

Discrete Lagrange-Multiplier Theory (cont'd)

- *Saddle point* (SP_{dn}) $(x^\star, \lambda^\star)$

$$L_d(x^*, \lambda) \le L_d(x^*, \lambda^*) \le L_d(x, \lambda^*) \tag{7}$$

  for all $x \in \mathcal{N}(x^*)$ and all $\lambda \in R$.

- *Constrained local minimum* (CLM_{dn}):
  - Feasible and $f(x') \ge f(x)$ for all $x' \in \mathcal{N}(x)$
- *Constrained global minimum* (CGM_{dn}):
  - Feasible and $f(x') \ge f(x)$ for all x' in search space
- First-order necessary and sufficient conditions for CLM_{dn}
  - One-to-one correspondence between CLM_{dn} and SP_{dn}

Constrained Simulated Annealing (CSA)

- Look for discrete-space saddle points by doing descents in free-variable w space and ascents in Lagrange-multiplier space
- Converge to CGM_{dn} under certain conditions

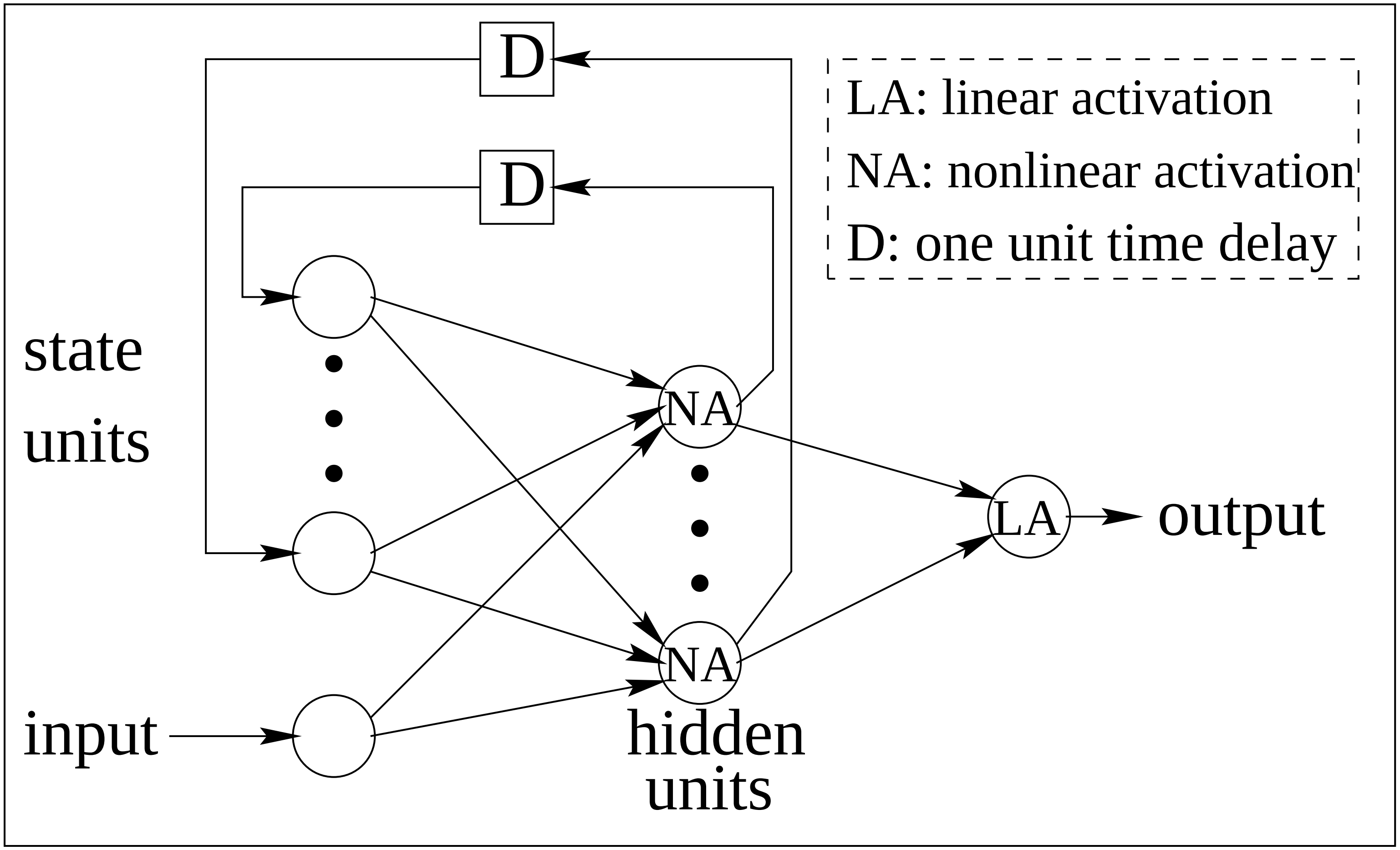
procedure CSA
    set initial $\mathbf{x} = (x, \lambda)$ by randomly generating x and by setting $\lambda \leftarrow 0$;
    initialize starting temperature T to be large enough and the cooling rate $0 < \alpha < 1$
    set N_T (number of trials per temperature);
    while stopping condition is not satisfied **do**
        for $n \leftarrow 1$ **to** N_T **do**
            generate $\mathbf{x}'$ from $\mathcal{N}(\mathbf{x})$ using $G(\mathbf{x}, \mathbf{x}')$;
            accept $\mathbf{x}'$ with probability $A_T(\mathbf{x}, \mathbf{x}')$
        end_for
        reduce temperature by $T \leftarrow \alpha \times T$;
    end_while
end_procedure

Integration of Back-Propagation and CSA

- Random sampling too expensive
  - Use epoch-wise back-propagation through time (EWBPTT)
- Integration of EWBPTT into CSA to generate new try point $w + \delta w$

Architecture in Use

- Recurrent ANN: 1 input unit, n hidden units, and 1 output unit



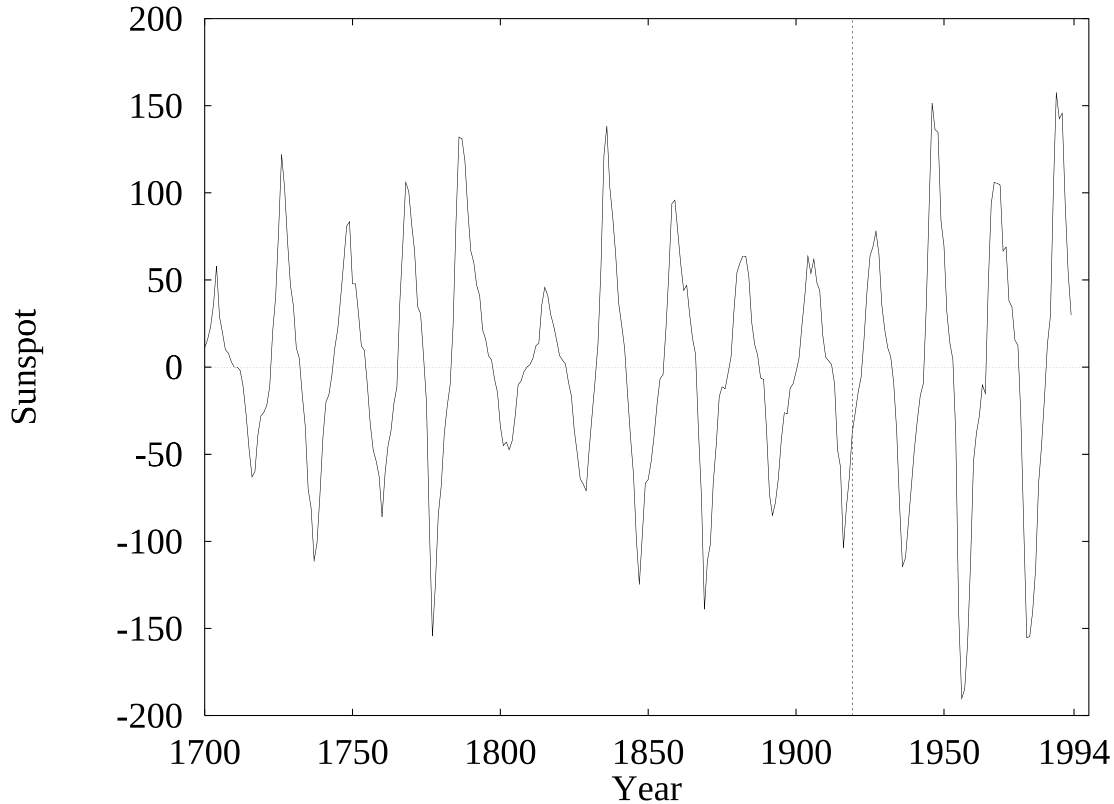
Comparisons of 5 Training Methods

SA:	Unconstrained,	SA,	No cross-validation
SA&V1:	Unconstrained,	SA,	Traditional cross-validation
CSA:	Constrained,	CSA,	No cross-validation
CSA&V1:	Constrained,	CSA,	Traditional cross-validation
CSA&V2:	Constrained,	CSA,	Proposed cross-validation

Test Data:

- Sunspot time series: yearly number of sun-spots (1700-1994)
  - Training data: 1700-1920
  - Testing data: 1921-1994
  - Cross-validation data: 1900-1921

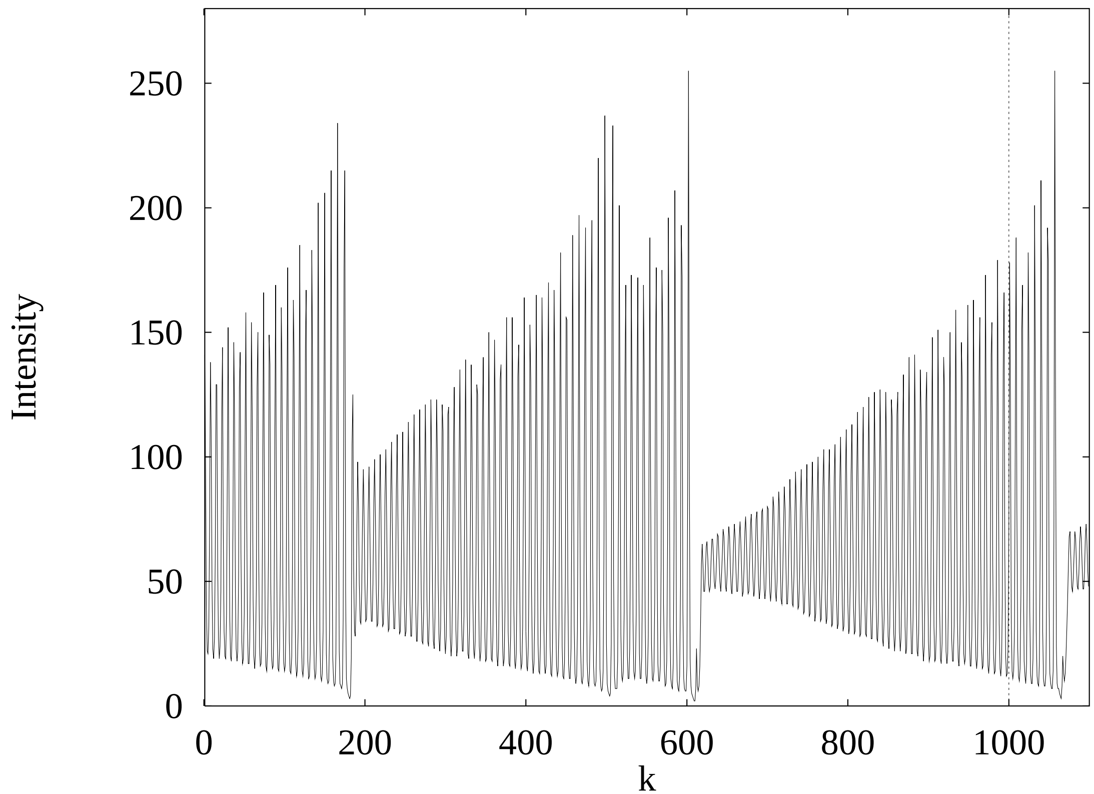
Sunspots Time Series



Dotted vertical line separates training set and test set

- ANN used: 1 input, 2 hidden units, 1 output – 11 weights in total

Laser Time Series



Dotted vertical line separates training set and test set

- Laser time-series: 1 input – 20 hidden units – 1 output, 461 weights

Results on Sunspots

- 24 to 26 seconds per run on a 450MHz P III with Solaris 2.7

Average prediction nMSE (with 95% confidence on $\pm 10\%$)

Method	1921-1955	1956-1979	1980-1994	1921-1994	Runs
SA	0.052306	0.113958	0.055074	0.076735	28
SA&V1	0.081943	0.138510	0.086798	0.102199	76
CSA	0.035385	0.061361	0.039559	0.045554	10
CSA&V1	0.042079	0.086607	0.051244	0.060784	18
CSA&V2	0.034288	0.053549	0.034236	0.040634	4

- Observations:
  - Without cross-validation, CSA consistently out-performs SA
  - Traditional cross-validation does not work well
  - CSA&V2 out-performs all other algorithms and is the most stable

Comparison with Previous Work on Sunspots

Method	No. of Free Variables	Training	Single-Step Testing			
		1700-1920	1921-55	1956-79	1980-94	1921-94
AR(12)	14	0.128	0.126	0.36	0.306	0.238
TAR	18	0.097	0.097	0.28	0.306	0.197
WNet	113	0.082	0.086	0.35	0.313	0.219
SSNet	N/A	-	0.077	N/A	N/A	N/A
DRNN	30	0.105	0.091	0.273	N/A	N/A
COMM	N/A	0.079	0.065	0.24	0.188	0.148
ScaleNet	N/A	0.086	0.057	0.13	N/A	N/A
Proposed CSA&V2	11	0.0559	0.0337	0.0524	0.0332	0.0397

AR(12): 12^{th}-order linear auto-regression

TAR: Threshold auto-regressive model

WNet: Feedforward ANN with weight elimination

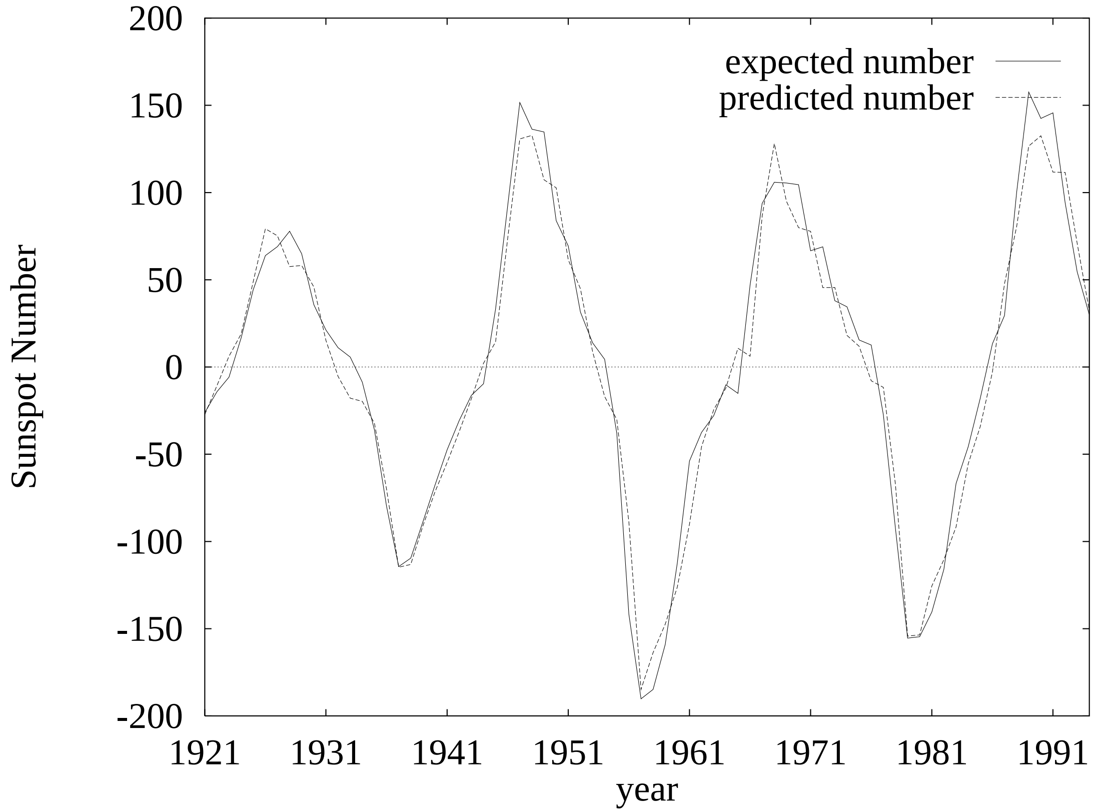
SSNet:Soft-weight-sharing network

DRNN: Dynamic recurrent ANN

COMM: Committee prediction using FIR network

ScaleNet: Multiscale ANN

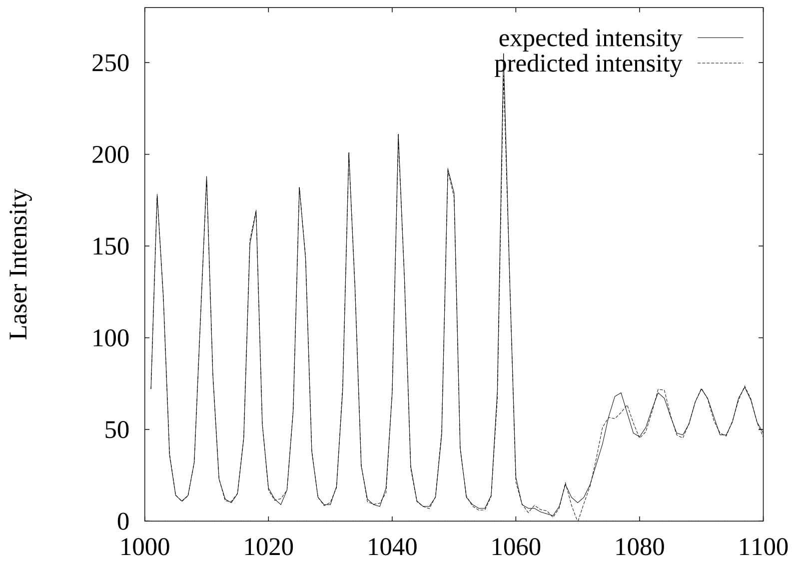
Single-step Prediction for Sunspots



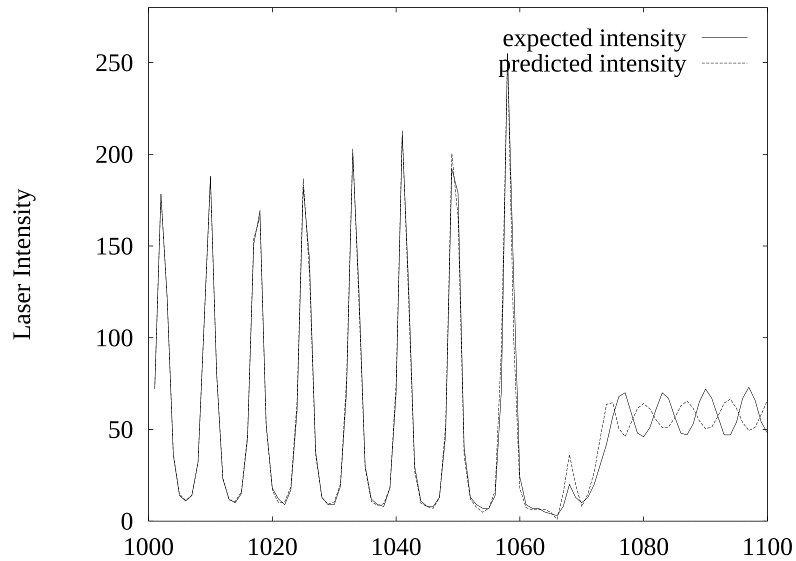
Comparisons with Previous Work on Laser

Method	Number of weights	Training	Single Step Prediction		Iterative Prediction	
		100-1000	1001-1050	1001-1100	1001-1050	1001-1100
FIR network	1105	0.00044	0.00061	0.023	0.0032	0.0434
ScaleNet	N/A	0.00074	0.00437	0.0035	N/A	N/A
CSA&V2 (Run 1)	461	0.00036	0.00043	0.0034	0.0054	0.0194
CSA&V2 (Run 2)	461	0.00107	0.00030	0.00276	0.0030	0.0294

Predictions for Laser Time-Series



Single-step prediction



Iterative prediction

Conclusions

- Constrained formulation with proposed cross-validation method CSA$V2 is effective, stable, and out-performs previous work significantly on two benchmarks
- Future work
  - Improve speed and solution quality of training algorithm
  - Study non-stationary time-series: stock and currency exchange time-series